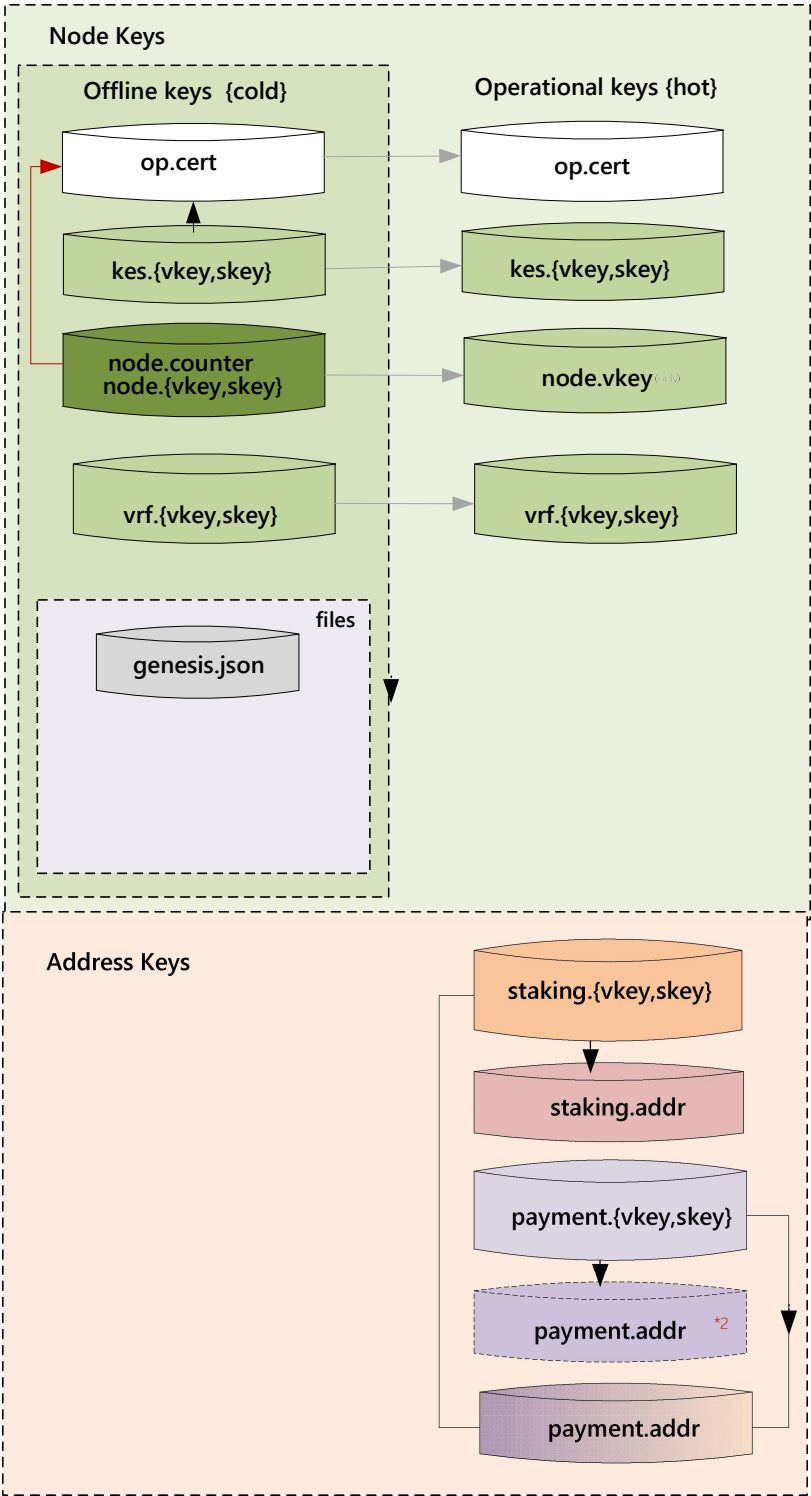


# Node Keys and Operational Certificate

cardano-cli shelley

node	key-gen	--verification-key-file --signing-key-file --operational-certificate-issue-counter
	key-gen-KES	--verification-key-file --signing-key-file
	issue-op-cert	--kes-verification-key-file --cold-signing-key-file --operational-certificate-issue-counter --kes-period --out-file
	key-gen-VRF	--verification-key-file --signing-key-file

stake-address	key-gen	--verification-key-file --signing-key-file
	build	--staking-verification-key-file <i>staking.vkey</i> > <i>staking.addr</i>
address	key-gen	--verification-key-file --signing-key-file
	build	*--payment-verification-key-file <i>payment.vkey</i> > <i>payment.addr</i>
	build	--payment-verification-key-file <i>payment.vkey</i> --staking-verification-key-file <i>staking.vkey</i> > <i>payment.addr</i>



*kes.vkey*  
5 2 45d4...ca89  
8 0

*node.vkey*  
5 2 f4bb...1ff7  
8 0

*vrf.vkey*  
5 2 0000...2a83  
8 0

*staking.vkey*  
1 b 5 2 c353...b7e9  
8 9 8 0

Stake address  
5 2 e ccdf...330d  
8 1 0

*payment.vkey*  
1 a 5 2 8361...63e5  
8 f 8 0

Enterprise address  
6 0 2ffd...570e

Base address  
0 0 payment.addr staking.addr 0 0 2ffd...570e ccdf...330d

wallet 6 0 6466...87cf

Faucet

\*2 This is not the payment.addr you are looking for. Use the base payment.addr combo to stake and use payment.{vkey,skey} to witness the payment input of transactions.

\* export CARDANO\_NODE\_SOCKET\_PATH=\$PWD/example/core/sockets/node.socket

\* --testnet-magic 42 or --mainnet